Tablexcel: A Multi-user, Multi-touch Interactive Tabletop Interface for Microsoft Excel Spreadsheets

Guillaume Besacier

Tablexcel: a multi-user, multi-touch interactive tabletop interface for Microsoft Excel spreadsheets

Guillaume Besacier[1, 2]

[1] LIMSI-CNRS, BP 133, 91403 Orsay Cedex, France
[2] Université Paris-Sud, 91405 Orsay Cedex, France
guillaume.besacier@limsi.fr

Abstract. In this paper, we present Tablexcel, a tabletop interface to Microsoft Excel. Single-user, desktop-based computer applications are pervasive in our daily lives and work. An application like Microsoft Excel, a widely deployed spreadsheet application, is used by a large number of businesses and users. Often, several users will collaborate on the creation of a spreadsheet, for example exchanging Excel files by e-mail. A multi-user, multi-touch interactive tabletop could create better working conditions, but Excel is not compatible with tabletop interfaces. Tablexcel use the scripting capabilities of Excel to extract live data from Excel files, and display them in a tabletop-appropriate way. Multiple users can interact with the Tablexcel interface using tabletop interactions, like gestures or rotating windows. Tablexcel manage the collaborative aspect of the interaction and send the resulting modifications to the original Excel application, which update the formulas, graphs, macros, etc.

Keywords: Interactive tabletop, legacy application, spreadsheet, scripting.

1 Introduction

Single-user, desktop-based computer applications are pervasive in our daily lives and work. Being able to use these applications with novel interactive systems, like multi-touch tabletops, is a requirement for a large deployment of these systems in a production environment. Indeed, these applications, whether they are widely used or specialized business applications, are essential to their users.

Microsoft Excel, a widely deployed spreadsheet application, is used by a large number of businesses, either for its own value or as a platform for third-party applications programmed with its macro language. Often, several users collaborate to create a single spreadsheet. This author had received and sent back Excel file by e-mail for such diverse purpose as cooperatively making a planning, drafting a list, or filing forms. Excel files are also edited during face-to-face meeting, for example to tweak the weight given to the various grades during an academic jury. Often, a meeting leader will modify the master file on his or her laptop, while the other participants try to follow and make the same modification on their own laptop. A multi-user, multi-touch tabletop could create better working conditions in both situations.

Researchers have created new applications, designed for interactive tabletops, to fill the role of existing applications. For example, a spreadsheet application [4] was programmed with the T3 tabletop toolkit. But it is not compatible with existing file formats, and offer only basic functionalities. In addition, it doesn't offer a rich, collaborative multi-user experience, as it uses an off-the-shelf single-user component.

This lack of real-world applications limits the appeal of tabletop computing for businesses and end-users, but it also limits our ability, as HCI researchers, to conduct long term studies and field studies on tabletop usage. The only such study we are aware of was conducted using a single user in a desktop environment using a tabletop to emulate a mouse [5].

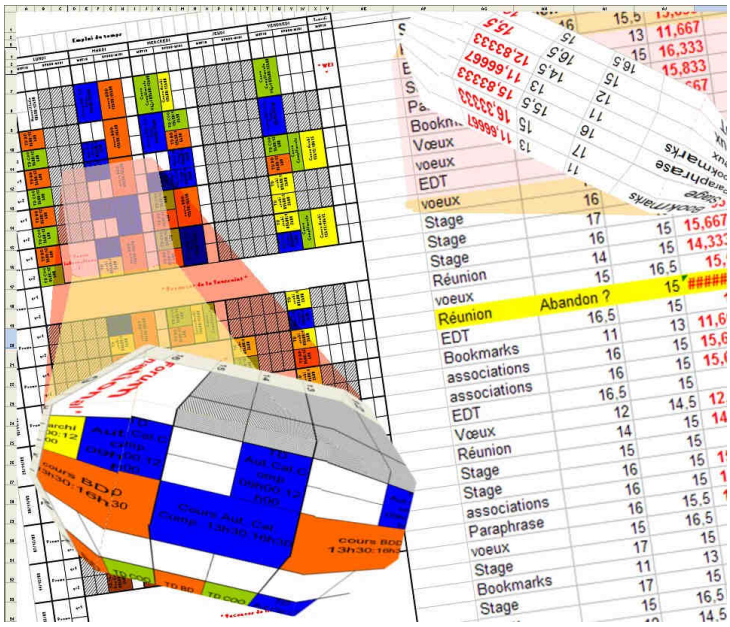

Fig. 1. Prototype of the Tablexcel user interface. Each user can have their own focus point.

2 Implementation

Several technologies have been indentified to use existing applications with innovative interactive systems [1]. In this paper, we describe our implementation of Tablexcel, a tabletop interface to Microsoft Excel. Excel exposes a scripting interface. Its purpose is to facilitate tasks automation and applications interactions. For example, a third-party a word processing application might ask a spreadsheet application for some data to create a mailing. Scripting allows Tablexcel to access Excel raw data without formatting for a particular viewport, with the burden of presenting the data in a visual way delegated to our interface.

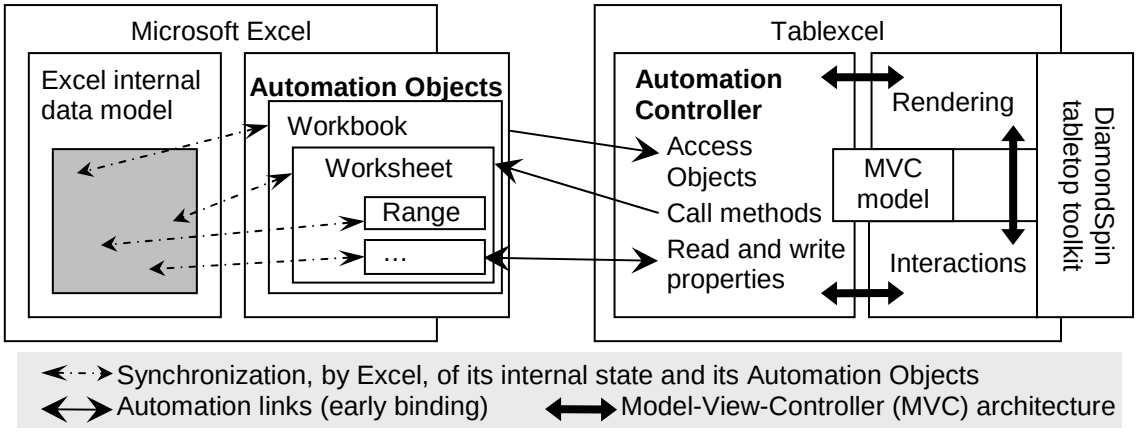

Fig. 2. Tablexcel MVC architecture delegates the Model part to the actual Excel application.

Excel exposes its data model with Automation objects. Automation is an inter-process communication mechanism based on the Microsoft Component Object Model (COM). It provides an infrastructure whereby applications called Automation Controllers can access and manipulate (i.e. set properties of or call methods on) shared Automation Objects that are exported by other applications [2]. The Excel Automation Object Workbook represents a file. It contains Worksheet objects, which contain Range objects representing a cell. Each cell has several properties: the numeric value of the cell, the formula used to compute it, its size, font, color, etc.

We implemented an Automation Controller, called Tablexcel, in C++ and Java, which uses the Automation Objects exported by Excel as its data model (Figure 2). Tablexcel uses the Model-View-Controller (MVC) architecture. The Model is composed of Automation Objects exported by Excel. Excel synchronizes theses objects with its internal state. The Controller and View are implemented using the DiamondSpin tabletop toolkit [3]. For maximum speed, the rendering is done in OpenGL using a custom algorithm created specifically to display Excel spreadsheets.

3 Interacting with Tablexcel

Tabletop spreadsheets prototypes developed so far have retained the desktop metaphor, using windows, scrollbar and similar widgets to interact with their users. We think this metaphor is not the most efficient when several users are working on the same spreadsheet. Our inspirations for Tablexcel were the interactions one can have with a large map (which is an area frequently studied on tabletops).

The spreadsheet is displayed on the whole table. Is can be translated and rotated with a finger touch, and zoomed with a pinch gesture. The translation and rotation use a kinetic relation between the finger movement and the spreadsheet movement rather than a direct mapping: the movement starts slowly and then pick up speed. As the spreadsheet is shared by all the users, it can disturb a user work if another user

suddenly moves it. A slow initial speed allows for easy cancellation of the movement in its initial phase. A user can cancel the movement by putting their whole hand on the table to metaphorically hold the spreadsheet in place.

This view of the spreadsheet cannot be modified. The size and resolution of most current tabletops doesn't allow small enough widgets to interact directly with this view, which need to be broad enough that all the users' centers of interest fit in the tabletop surface. Instead, users can open an interactive secondary view (two such views are opened in Figure 1), by delimiting an area of interest between the side of their hands. These views are then managed like windows in other tabletop applications: they can be rotated, moved, zoomed, etc. Each user has their own focus, and a formula bar on the tabletop border next to them (not visible in Figure 1). The focused cell is editable using a physical wireless keyboard per user.

4 Conclusion

We presented Tablexcel, a multi-user, multi-touch tabletop interface to Microsoft Excel using the scripting capabilities of Excel to extract live data from Excel files. In future works, we plan to use Tablexcel to study users' interactions with an interactive tabletop in field studies using users' actual data and work processes.

References

1. Besacier, G., Vernier, F. Toward user interface virtualization: legacy applications and innovative interaction systems. Proc. EICS'09, pp. 157-166.
2. Kruglinski, D., Wingo, S., Shepherd, G. Automation. In Programming Microsoft Visual C++ 6.0 (5th ed.). Microsoft Press, 1998.
3. Shen, C., Vernier, F., Forlines, C., Ringel, M. DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction. Proc. CHI 2004, pp. 167-174.
4. Tuddenham, P., Robinson, P. T3: Rapid Prototyping of High-Resolution and Mixed-Presence Tabletop Applications. Proc. Tabletop'07, pp. 11-18.
5. Wigdor, D., Perm, G., Ryall, K., Esenther, A., Shen, C. Living with a Tabletop: Analysis and Observations of Long Term Office Use of a Multi-Touch Table. Proc. Tabletop'07, pp. 60-67.